ANALYSIS OF WEB LOG DATA USING APACHE PIG IN HADOOP

A. C. Priya Ranjani* & Dr. M. Sridhar**
*Research Scholar, Department of Computer Science, Acharya Nagarjuna University, Guntur, Andhra Pradesh, INDIA,
**Associate Professor, Department of Computer Applications, R.V.R & J.C College of Engineering, Guntur, India

Received: April 09, 2018 Accepted: May 22, 2018

ABSTRACT *The wide spread use of internet and increased web applications accelerate the rampant growth of web content. Every organization produces huge amount of data in different forms like text, audio, video etc., from multiplesources. The log data stored in web servers is a great source of knowledge. The real challenge for any organization is to understand the behavior of their customers. Analyzing such web log data will help the organizations to understand navigational patterns and interests of their users. As the logs are growing in size day by day, the existing database technologies face a bottleneck to process such massive unstructured data. Hadoop provides a best solution to this problem. Hadoop framework comes up with Hadoop Distributed File System, a reliable distributed storage for data and MapReduce, a distributed parallel processing for executing large volumes of complex data. Hadoop ecosystem constitutes of several other tools like Pig, Hive, Flume, Sqoop etc., for effective analysis of web log data. To write scripts in Map Reduce, one should acquire a good programming knowledge in Java. However Pig, a simple dataflow language can be easily used to analyze such data. This paper details the use of Pig Latin for discovering the hidden patterns in millions of web records.*

Keywords: *Web Usage Mining, Hadoop, Pig, MapReduce.*

Introduction

The number of users of internet is growing exponentially day by day. Massive volumes of data is being generated and accessed by millions of users all over the globe. The analysis of such web data called Web Mining is very crucial for organizations to upgrade their business and market value. Web mining is the implementation of data mining techniques [1][2] to retrieve, extract and analyze information from web data which incorporates web document pages, navigations between documents, usage of web sites etc.The expansion of the World Wide Web (WWW) has contributed to accumulation of large amount of data that is freely obtainable for access by any user. A log file keeps a registry of events, processes, messages and communications made between various software applications and the operating systems. Log file is the one which records every activity of executable files. Analyzing logs is significant to find the usage of a resource, to identify HTTP errors and slow queries, to give quick response to security threats, to track your site visitors. Log data analysis will disclose sensitive information about the attackers.

Web Content Mining, Web Structure Mining and Web Usage Mining are the three main categories that fall under Web Mining. Content Mining is the process of eliciting many interesting facts from the content of web documents. According to kosala and Blockeel [3] it is the extraction of useful knowledge from web content, data and documents. Structure Mining is performed on hyperlink structures and it focuses on refining the structural design of a website. Usage mining discovers the usage patterns of data stored in web logs. It's main emphasis is on finding information about the navigational patterns and behavior of the user[4][5].

Our paper mainly concentrates on Web Usage Mining. The layout of the paper is as below. Section I introduces web mining, describes its significance, and explains its methodology in brief. Section II details on Web Usage Mining. Section III discusses the use of Hadoop framework and its tools in web usage mining and Section IV describes Pig architecture and illustrates the application of Pig queries to analyze data and records the corresponding results. Finally section V describes the conclusion and further enhancements of our work.

Web Usage Mining

Web Usage Mining (WUM) has gained much attention now-a-days. The identity or origin of web users along with their browsing behaviour[6] at a web site can be captured with the aid of WUM. The uncovered patterns are then applied to enhance the design of a website [7], web personalization , web caching and making additional topic or product recommendations.The log data can be collected at server level, client level and proxy level. A web server keeps track of four types of log files: Transfer log, Referrer

log, Access log and Error log. Of these, the referrer and agent logs may or may not be started at the server. If required they can be added to the transfer log file to create an extended log file format. The information about the web pages requested by users is kept inside access logs. The error log stores all requests that were failed and the reasons for the failure.

The log files are automatically created by web server on every hit or request to the web site by the user. The web server retains information like who are visiting the website, what activities they are performing, how they are directed to that page, what is the status of the request, what amount of data is transferred etc. This data is basically in raw form and needs preprocessing. There were several standard formats like Common Log Format, Combined Log Format, and Extended Log Format to preserve the data about user activities.

A. Log Formats

The structure of the Common Log Format is given below,

"%h %l %u %t \"%r\" %>s %b"

Example of a common log format is

133.43.96.45–adams[01/Aug/1995:00:00:23-0400]"GET/images/launch-logo.gifHTTP/1.0"2001713

- 133.43.96.45 (%h) -IP address of the client who made request to the server
- (%l) - Hyphen indicates that the requested data is not available
- adams(%u) - The user id of the person requesting the document
- [01/Aug/1995:00:00:23 -0400] (%t) -The timestamp resembles [day/month/year: hour: minute: second zone]
- "GET /images/launch-logo.gif HTTP/1.0" - The request from the client in double quotes, HTTP/1.0 is the protocol version.
- 200 (%>s) - The status code sent by the server.
- 1713 (%b) - The number of bytes sent to client by the server

The configuration of the Combined Log Format is shown below,

"%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\" \"%{User-agent}i\""

The entries in CLF log file will look as below :

133.43.96.45 – adams [01/Aug/1995:00:00:23 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713"http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

The extra parameters included in the Combined Log format are discussed below

- "http://www.example.com/start.html" (\"%{Referer}i\") - This names the site that the client reports having been referred from.
- "Mozilla/4.08 [en] (Win98; I ;Nav)" (\"%{User-agent}i\") - This is the data sent by the client browser about itself to the server.

Extended Log Format (ELF): This format includes additional information such as the referring URL, name and version of the browser, and the operating system of the host. This log format covers W3C standard, supported by Apache and Netscape web servers. It also includes W3SVC standard, supported by Microsoft Internet Information servers. The following is an example file in the extended log format.

#Version: 1.0
#Date: 18-Feb-2010 00:00:00
#Fields: time cs-method cs-uri
00:34:23 GET /base/xyz.html
12:21:16 GET /base/xyz.html
12:45:52 GET /base/xyz.html

12:57:34 GET /base/xyz.html
Remark: <text>

- version: <integer>.<integer> - The version of the ELF used which is 1.0 here.
- Fields: [<specifier>...] - Lists the fields included in the log.
- Software: string - Identifies the software which generated the log.
- Start-Date: <date> <time> - The date and time at which the log was initiated.
- End-Date:<date> <time> - The date and time at which the log was completed.
- Date:<date> <time> - The date and time at which the entry was added.

B. States of Web Usage Mining

There are three prominent phases in WUM – Preprocessing, Pattern Discovery and Pattern Analysis as shown in figure 1. The data gathered from sources like server, client and proxy first needs to be preprocessed.

1)Data Preprocessing : It refers to the activities that are executed to make the raw data suitable for further processing or analysis [8][9]. Data from various sources is integrated and it should be consistent. Preprocessing includes removal of unwanted data, session identification, user identification and path completion.

2)Pattern Discovery : In this stage, preprocessed data undergoes rigorous analysis to discover the valuable patterns[10], Statistical techniques and machine learning method scan be used to mine the patterns. The frequently used approaches are: association rules, clustering, classification, path analysis, sequential patterns and model discovery.

3) Pattern Analysis : After the discovery of patterns, techniques and tools are required to make these patterns understandable for analysts and to utilize them at the maximum. Interesting patterns from several available patterns should be selected. Techniques for pattern analysis include database querying, graphics and visualization, statistics and usability analysis.

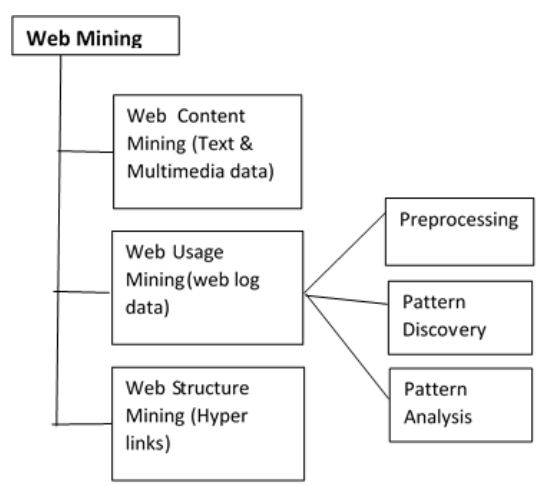


Figure 1. Types of Web Mining

HADOOP FRAMEWORK

Large websites that handle millions of simultaneous visitors may generate hundreds of petabytes of logs per day. The traditional RDBMS cannot be used for storage and retrieval ofmassive heterogeneous datasets. For efficient and effective analysis of such big weblogs, we need to establish parallel, scalable and faster data mining techniques. Also there is a need for cluster of storage devices and a parallel computing model for loading and examining the data. Hadoop framework satisfies these needs through HDFS (Hadoop Distributed File System) for distributed storage of large datasets and a MapReduce model for parallel processing. Besides these other tools like Sqoop, Flume, Pig, Hive are also available for handling the massive and complex datasets.

For productive analysis of weblogs, many researchers have applied Hadoop based tools. Vernekar and Buchade [11] in 2013 proposed the use Hadoop Map Reduce on log analysis for identifying problem and system threats. In 2014, the authors [12] have designed and implemented an enterprise weblog analysis system based on Hadoop Map Reduce and Pig Latin language. The system was successful in identifying potential problems and predict the future trends of an application server traffic. J.Nandimath et al. [13] have proposed a scheme to analyze big data using Apache Hadoop. The data is stored in Mongo DB, which is a NoSQL database and MapReduce is applied for processing the data. Therdphapiyanak and Piromsopa [14] used Hadoop for analysis of Apache web server logs. They used distributed K-Means algorithm based on Mahout and MapReduce model and achieved a better performance compared to standalone log analyzer. Hingave and Ingle [15] also proposed a log analyzer with the combination of Hadoop and MapReduce paradigm.

A. MapReduce Framework

Parallel processing is the technique used for efficient processing of large and complex datasets. It addresses issues concerned with storage and execution of huge volumes of data. A number of parallel processing techniques have emerged over time but the ultimate option to overcome recent challenges is provided by Hadoop. MapReduce [16] is a simple programming model of Hadoop for parallel execution of data. The MapReduce code can be implemented in different languages like Java , Ruby and Python. These programs are inherently parallel in nature and can be run over a cluster of nodes. There are two prominent tasks in a MapReduce. One is Map Task and the second one is Reduce task. The large input file is partitioned into multiple splits. For each split, a separate map task is initiated. The input to map task is list of key-value pairs and the output is also a list of intermediate key-value pairs. The output from Map task is supplied as input to Reduce task. The reduce task accepts intermediate key-value pairs and generates a reduced and summarized list of key-value pairs. The log files with thousands and millions of records in the format of text are partitioned and given as input to several Map tasks and will be run concurrently. In business

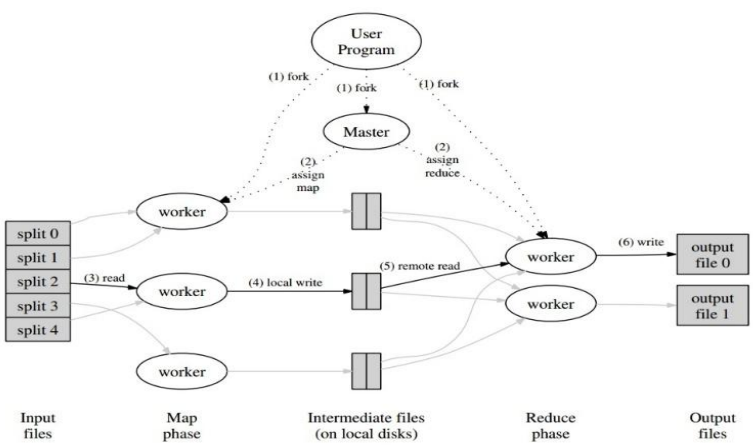


Figure 2. Map Reduce scenario for processing a job

perspective, there is a need to process these log files [17] so that we can have a view on the current trends in ongoing business. Each log record is read from an input text file, breaks it into a sequence of keys (x1, x2, …., xn) and sets value for each key as 1. If a key appears n number of times among all records then there will be n pairs of the form (x,1) among its output. Map: (x1, v1) -> [(x2, v2)]. Reduce task takes key and its list of associated values as an input. It merges values for same input key and reduces list into a single value. This single value is the count of occurrences of each key in the log file, and output is in the form of (x, n). Reduce : (x2, [v2]) -> (x3, v3).

B. Sqoop& Flume

Apache Sqoop has been designed to work with structured data. It is able to connect with RDBMS and move the structured data into the HDFS. The JDBC connector in Sqoop allows to move data out of different kinds of relational database services such as MySQL, SQL+, SQLServer into HDFS. Apache Flume has been designed for streaming data. It is a reliable distributed service that collects, aggregates and moving large amounts of streaming data into the Hadoop Distributed File System . For instance, it will collect log files

from multiple web servers and then move the log events from those files into an aggregated file in HDFS. It's simple and flexible architecture for streaming data flows makes it robust and fault tolerant. It has best reliability mechanisms for failover and recovery.It can be used not only for log data aggregation, but also to transport network traffic data, data from social websites, email messages etc.

C. HIVE

Apache Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analysis easy. It is specifically designed for OLAP operations. Hive provides an SQL-like interfacecalled HiveQL, to query data stored in various databases and file systems that integrate with Hadoop. It facilitates reading, writing, and managing large datasets residing in HDFS using SQL. It provides Hive Web User Interface and a Command Line Interface for user connectivity.

Hive is best suited for traditional data warehousing tasks such as extract/transform/load (ETL), reporting and data analysis rather than online transaction processing (OLTP) tasks. Hive is designed to boost up scalability, performance, extensibility, robustness and loose-coupling with its input data using variety of data formats. It can provide access to all files available in HDFS or storage platforms such as HBase.

D. Apache Pig

Pig an open source high-level data flow system, is a layer of abstraction built on the top of Map Reduce. It can do better on any data size, type or location. Pig has attained popularity since its introduction in 2006 by Yahoo. It promotes an adhoc way of creating and executing Map Reduce jobs on very large data sets. It is better than all RDBMS and DBMS based on performance, storage, and transaction level fault tolerance. Apache Pig has been proven to be the most efficient tool for analyzing structured, semistructured and unstructured data. Pig uses ETL process for data warehousing. ETL, which stands for "Extract, Transform and Load" is the set of functions combined in one tool to extract large amount of data from numerous databases. Pig furnishes a high-level dataflow language called Pig Latin. It has several operators, that help programmers in developing their own functional units for reading input, writing output, and processing the data. Pig queries and functions can be easily translated into a series of MapReduce jobs which are then run on a Hadoop cluster. Performance, time complexity and the accuracy of data is vital for any enterprise and this can be accomplished through Apache Pig.

PROCESSING DATA WITH PIG

A. Pig Architecture

Apache Pig simplifies the processing of large datasets. Writing MapReduce code requires intrinsic knowledge of Java along with sound programming skills. Even after writing the code, extra time is required for assessment of code. Pig supports complex, nested data structures which differentiates it from SQL supporting flatter data structures. It reduces the length of the code by using multi-query approach. The work done by a 200 lines Java program can be replaced by just 10 lines of Pig code. Although Pig scripts are 50% slower in execution compared to MR programs, they still are very effective in increasing productivity of data engineers and analysts by saving lots of time in writing phase. Pig is made up of two components Pig Latin language and the execution environment to Pig programs. The architecture for Pig is shown in the figure3. The major constituents of Apache Pig are as mentioned:

Parser: Initially, the Pig Scripts are handled by the Parser. The script is verified for syntax, data types and other miscellaneous details. The output of the parser can be depicted as a DAG (directed acyclic graph). In the DAG, the logical operators of the script are labelled as nodes and the data flows as edges.

Optimizer: The logical plan (DAG) is produced for each line of the script after semantic checking and basic parsing. This is sent to the logical optimizer, which performs the logical optimizations like projection and pushdown.

Compiler: The compiler translates the optimized logical plan into a chain of MapReduce jobs.

Execution Engine: Eventually the execution engine yields the series of MapReduce jobs to Hadoop in a sorted manner. These MapReduce jobs are then implemented on HDFS to produce the desired result.

Pig has an enriched collection of operators [18] to perform various operations such as sort, filter, join, group, co group etc. Since it is similar to SQL, one can easily write Pig scripts. It provides immense support to specify custom processing in the form of user defined functions (UDFs). Languages

like Java, Python, and JavaScript can be used to implement Pig UDF's. Pig also facilitates Piggy Bank, a repository for Java UDFs and provides an interface to access Java UDFs written by other users and contribute our own UDFs for use by others. A pig script can be processed in local mode as well as Map Reduce mode.

Local Mode : We can choose this mode when we want to parse all input files from the local machine and the local file system. This is very simple and easy to use since there is no involvement of HDFS. This is the best option suitable for initial error detection and correction . To start pig in local mode use,

$ pig –X local

grunt> run '/home/User1/pig/query1.pig'

MapReduce mode: The second way of invoking grunt shell is the MapReduce mode. In order to execute the data residing in HDFS, we need to start Pig in MapReduce mode. The execution of a Pig script iimplicitly starts a MapReduce job in the background to perform specific operations on the input data.

$ pig –X mapreduce hdfs://localhost:9000/pig/Script1.pig

There three ways to implement Pig scripts – Interactive mode, Batch mode and Embedded mode.

Interactive Mode (Grunt shell) – The Grunt shell permits to run Apache Pig in interactive mode. In Grunt shell, the Pig Latin statements are executed one by one and eventually the output is triggered using Dump operator. We can use run and exec commands to view the output of a Pig script.

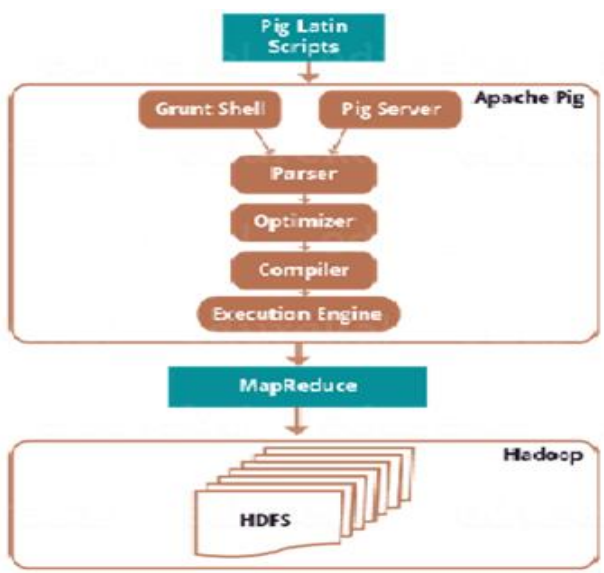


Figure 3. Architecture of Pig

Batch Mode (Script) – A number of Pig Latin statements and commands can be grouped and placed in a single file with .pig extension. This file called Pig script, can be run without the intervention of user.

EmbeddedMode (UDF) – Since Pig is an efficient data flow language, it allows the user to create his own methods or functions in programming languages like Java, Ruby and Python. These User Defined Functions (UDF's) can be easily embedded in any pig script.

Results & Discussion

The log data used in analysis are the one month's HTTP requests to the NASA Kennedy Space Center WWW server in Florida . The log reports contains logs collected from August 1, 1995, through August 31, 1995, a total of 31 days. The log file is defined in common log format. It contains Server IP

address, Timestamp, Method (GET / POST), Request URI (request_link andrequest destination) , HTTP status code, and Bytes accessed.The input file has 15,67,209 records. The logs are analyzed using Apache Pig commands. A sample of web server log file is shown below.

```
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:21 -
0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
133.43.96.45 - - [01/Aug/1995:00:00:22 -0400] "GET
/images/KSC-logosmall.gif HTTP/1.0" 200 1204
133.43.96.45 - - [01/Aug/1995:00:00:23 -0400] "GET
/shuttle/missions/sts-69/sts-69-patch-small.gif
HTTP/1.0" 200 8083
133.43.96.45 - - [01/Aug/1995:00:00:23 -0400] "GET
/images/launch-logo.gif HTTP/1.0" 200 1713
www-c8.proxy.aol.com - - [01/Aug/1995:00:00:24 -0400]
"GET /shuttle/countdown/ HTTP/1.0" 200 4324
133.43.96.45 - - [01/Aug/1995:00:00:25 -0400] "GET
/history/apollo/images/apollo-logo1.gif HTTP/1.0" 200
1173
ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:25 -
0400] "GET /shuttle/resources/orbiters/discovery-
logo.gif HTTP/1.0" 200 4179
piweba4y.prodigy.com - - [01/Aug/1995:00:00:32 -0400]
"GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
slppp6.intermind.net - - [01/Aug/1995:00:00:32 -0400]
"GET /history/skylab/skylab-1.html HTTP/1.0" 200 1659
```

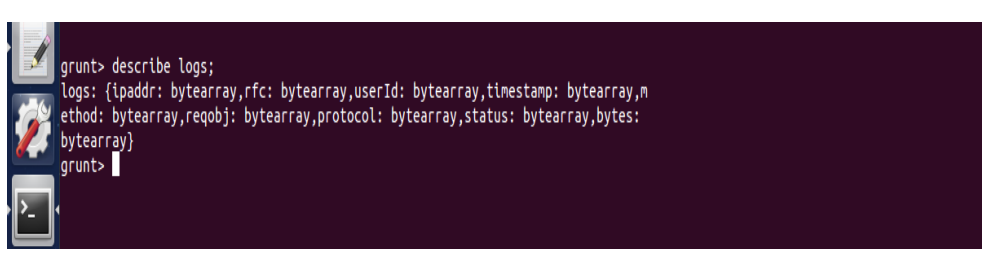
Figure 4. A snapshot of web log data

In the following section we have used Pig operators like LOAD, STORE, DUMP, GROUP, FILTER, SORT, FOREACH. To find no of visitors, most viewed websites, successful requests etc. We have also used Apache Common Log Loader defined in piggybank.jar. The sample results of our proposed work are shown below. We have installed Hadoop on a two machines with 4GB RAM and 1TB hard disk. Each machine having Ubuntu 14.04.

We have also installed Pig-0.11.0 for query processing. The log data is distributed evenly on these nodes and the Pig queries are run. For better understanding, this data is also shown in tabular form and graphical format. Table I displays the summarized report ofour analysis. Figures 6 and 7 exhibit the most viewed web pages and bandwidth consumed on daily basis respectively.

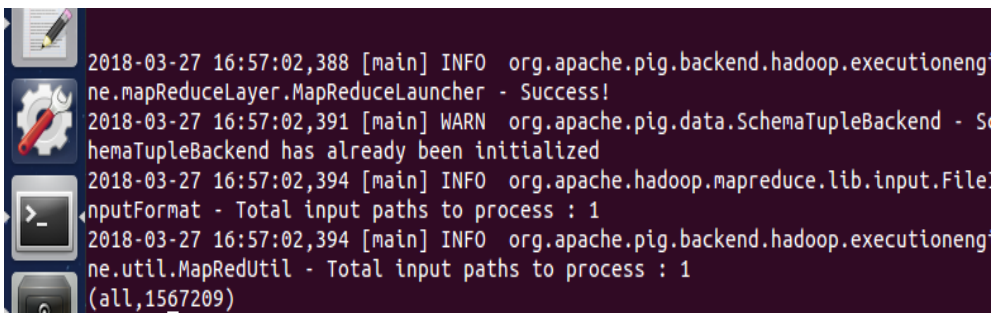
Query 1 : Load the data into a relation

```
REGISTER /home/user/acpr/pig-0.11.0/contrib/piggybank/java/piggybank.jar;
DEFINE ApacheCommonLogLoader org.apache.pig.piggybank.storage.apachelog. CommonLogLoader ();
DEFINE DayExtractor org.apache.pig.piggybank.evaluation.util.apachelogparser.
DateExtractor ('yyyy-MM-dd');
Mylogs = LOAD '/home/user/acpr/logdata' USING ApacheCommonLogLoader AS
(ipaddr,rfc,userId,timestamp,method,reqobj,protocol,status, bytes);
```



Query 2 : Total no of records in the relation

tot_recs = GROUP Mylogs ALL;
tot_recs_cnt = FOREACH tot_recs GENERATE group, COUNT (Mylogs);
DUMP tot_recs_cnt;

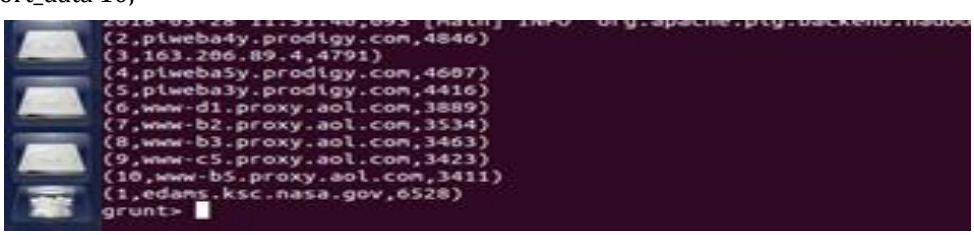


Query 3: Total number of request from a specific address

grpd = GROUP Mylogs BY ipaddr;
req_cnt = FOREACH grpd GENERATE group AS ipaddr, COUNT(Mylogs) AS tot_visits;

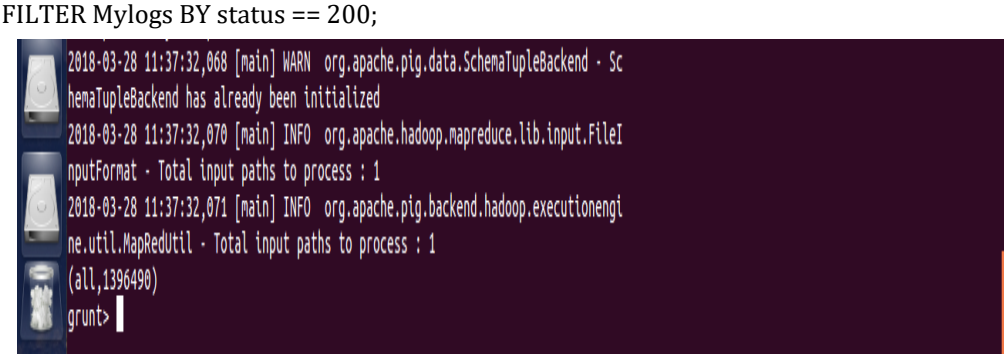
Query 4 : Rank the data based on total visits

sort_data = RANK req_cnt BY tot_visits DESC;
a = LIMIT sort_data 10;
DUMP a;



Query 5: Total number of successful requests

fil_logs = FILTER Mylogs BY status == 200;



Query 6: Display total bytes based on timestamp

```
time_data =  GROUP Mylogs BY DayExtractor(timestamp) as day;
byte_count = FOREACH time_data GENERATE group AS timestamp,SUM(Mylogs.bytes) AS total_bytes;
STORE byte_count INTO '/user/acpr/pig/ ApacheLogResult1' USING PigStorage(',');
```

Query 7 : Find unique hits to websites (IPs) per day

```
grpd = GROUP Mylogs BY DayExtractor(dt);
cntd = FOREACH grpdc
{            tempId =  Mylogs.ipaddr;
           uniqueUserId = DISTINCT tempId;
           GENERATE group AS day,COUNT(uniqueUserId) AS cnt; }
srtd = ORDER cntd BY cnt desc;
STORE srtd INTO '/user/acpr/pig/ ApacheLogResult2 ';
```

Table I Statistical Report

SUMMARY	
Total Hits	1,567,209
Visitor Hits	1,567,209
Total Visitors	143,765
Total Unique IP's	75,059
Total Bandwidth	24.19 GB
Visitor Bandwidth	24.19 GB
Successful Requests	1,396,490
Failed Requests	1,70,719
Average Bandwidth per Day	799.55 MB
Total Page Views	4,435
Average Bandwidth per Hit	19.07 KB

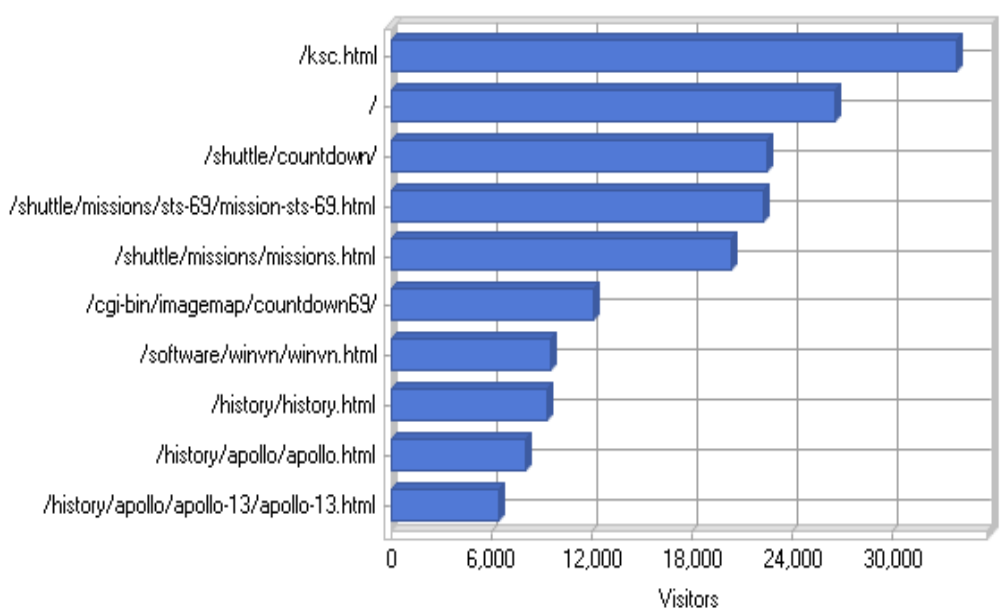


Figure 5. Most popular pages

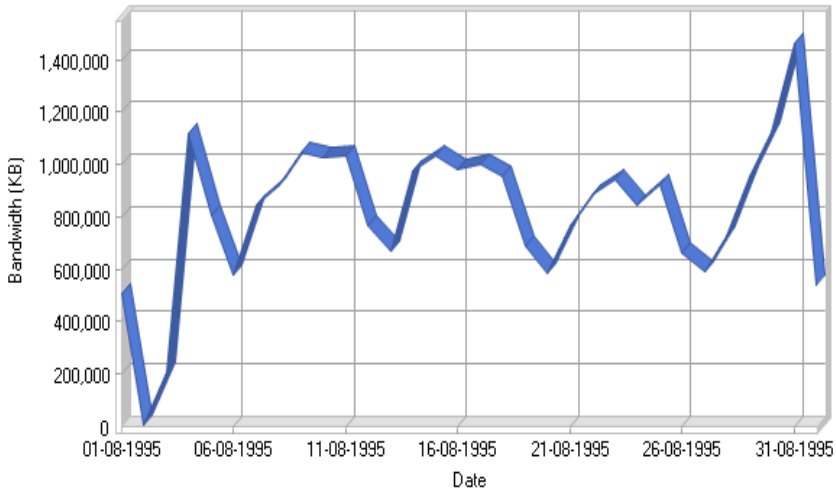


Figure 6. Bandwidth consumed daily

CONCLUSION AND FUTURE WORK

This paper gives a brief description of web usage mining process and application of Hadoop framework to discover the unseen interesting navigation patterns of the users. This information can be applied to find solutions to many legitimate problems such as web page/site improvement, additional product/topic recommendations, user/customer behavior analysis, etc. In this paper we have used Apache Pig which is an efficient and easy to use data flow language for analyzing the web logs. In future we plan to use big data clustering technique and map reduce paradigm for identifying user's access patterns.

REFERENCES

[1] Cooley, R.;Mobasher, B;&Srivastava, J.;"Web mining: information and pattern discovery on the world wide web";In Proceedings of 9th IEEE International Conference on Tools with Artificial Intelligence; 1997; pp. 558-567.

[2] Srivastava,J;Desikan, P;&Kumar V.; "Web Mining : accomplishments and future directions";Chapter 3; 2002.

[3] Kosala &Blockeel, H.;" Web Mining Research: A survey";SIGKDD Explorations Newsletter; ACM Vol.2 (1); 2000; pp.1-5.

[4] Thilagu, M. &Nadarajan R.;"Efficiently mining of effective web traversal patterns with average utility";In Proceedings of International Conference on Communication;Computing and Security;ELSEVIER; Vol.6;2012; pp. 444-451.

[5] Velayathan, Ganesan, Yamada &Seiji,"Behavior - based web page evaluation";In Proceedinds of IEEE/WIC/ACM International Conference on Web Intelligence&Intelligent Agent Technology Workshops; 2006.

[6] Yu-Shiang Hung, Kuei-Ling B. Chen, Chi-Ta Yang&GuangFeng Deng; "Web usage mining for analysing elder self-care behavior patterns";Expert Systems with Applications; Vol.40; No.2; pp. 775-783.

[7] Carmona, C.J.; Ramírez-Gallego, S.; Torres, F.; Bernal, E.;Del Jesus, M.J.; &García, S.;"Web usage mining to improve the design of an e-commerce website:OrOliveSur.com";Expert Systems with Applications; Vol.39; No.12; pp. 11243-11249.

[8] Suresh R.M.; Padmajavalli, R.; " An overview of data preprocessing in data and web usage mining";In Proceedings of First International Conference on Digital Management; Banglore; India; IEEE; 2006; pp.193-198.

[9] Joshila Grace, L.K.; Maheswari, V. &Nagamalai Dhinaharan; " Web log data analysis and mining"; InProceedings of CCSIT-2011, Springer CCIS, Vol.133, pp. 459-469.

[10] Pani, S.K.; Panigraphy, L.; Sankar. V.H.; BikramKeshariRatha, Mandal, A.K.; and S.K.Padhi, "Web usage mining: A survey on pattern extraction from web log", IJICA, Vol. 1, No.1, 2011.

[11] Vernekar, S.S.; Buchade, A;"MapReduce based log file analysis for system threats and problem identification"; In Proceedings of*IEEE 3rd International Advance Computing Conference (IACC)*; 2013; pp. 831-835.

[12] Wang; Chen Hau; Ching TsorngTsai; Chia Chen Fan & Shyan Ming Yuan; "A Hadoop based weblog analysis system"; In Proceedings of 7th International Conference on Ubi-Media Computing and Workshops (UMEDIA); 2014;pp. 72-77.